

Problem B. Раздел королевства

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 512 мегабайт

Правивший много лет король Байтландии Байтазар Большое Гнездо задумал приучить своих наследников к государственным делам. Для этого он собирается разделить своё королевство на герцогства и распределить между своими сыновьями. Проблема заключается в том, что детей у Байтазара много... реально много.

Для того, чтобы не провоцировать споров о старшинстве, Байтазар решил, что герцогства будут иметь попарно различную площадь. При этом некоторую часть младших наследников при необходимости можно оставить и без герцогства.

Королевство можно представить как прямоугольник $n \times m$, разбитый на $n \cdot m$ единичных квадратов. Каждое герцогство должно быть составлено из целого числа единичных квадратов и должно быть связным по стороне (то есть между каждыми двумя единичными квадратами, входящими в герцогство, существует путь, полностью составленный из входящих в герцогство единичных квадратов, в котором каждый следующий квадрат имеет с предыдущим общую сторону). Каждый квадрат должен принадлежать ровно одному герцогству.

Король планирует разделить королевство так, чтобы как можно большее количество наследников получило свои герцогства. Помогите ему сделать это.

Input

Единственная строка входа содержит два целых числа n и m ($1 \leq n, m \leq 1000$), обозначающие размеры прямоугольника.

Output

В первой строке выведите одно целое число k : наибольшее количество наследников, которые могут получить своё герцогство. Каждая из последующих n строк должна содержать по m символов — заглавных латинских букв из множества $\{A, \dots, Z\}$ — отображения герцогств. Отдельному герцогству соответствует связная часть, составленная из одинаковых букв и граничащая по сторонам только с частями, обозначенными другими буквами. Отметим, что различные части при этом вполне могут быть обозначены одной и той же буквой. При этом для любых двух частей количество составляющих эти части букв должно быть различно.

Гарантируется, что для любого соответствующего условиям задачи разбиения существует способ обозначить вышеописанным образом его части, используя не более чем 26 букв.

Examples

standard input	standard output
4 5	5 ABCCE BBCCE DDDEE DDEEE
1 12	4 AABCCCCAAAAA
8 3	6 CCC CCC ACB ACB ABB ABB CCC ABB

Problem C. Неквадраты

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 512 мегабайт

Задано целое положительное число n . Выясните, может ли оно быть представлено в виде произведения k целых положительных чисел, ни одно из которых *не* является квадратом целого числа.

Input

Первая строка ввода содержит целое число t ($1 \leq t \leq 10$) — количество тестовых случаев. Каждая из последующих t строк содержит один тестовый случай, состоящий из двух целых чисел n ($1 \leq n \leq 1\,000\,000\,000$) и k ($2 \leq k \leq 50$).

Output

Для каждого тестового случая выведите в отдельной строке слово «YES» в том случае, если существует такой набор из k положительных целых чисел a_i , что $n = a_1 \cdot a_2 \cdots a_k$ и ни одно из a_i не является квадратом целого числа, и слово «NO» в противном случае.

Examples

standard input	standard output
4	NO
1 2	YES
6 2	NO
7 2	YES
8 3	

Problem D. Fans Placement

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Фанатский сектор на футбольном стадионе города Б. представляет собой N рядов кресел, расположенных друг за другом на лестнице. Однако при постройке стадиона в целях предотвращения беспорядков было принято странное решение: в каждом ряду фанатского сектора поставили только одно кресло.

Известно, что на очередной футбольный матч в фанатский сектор хотят приобрести билеты K фанатов, причём для i -го фаната известен его рост h_i . Поскольку фанаты переживают за свою команду очень сильно, они собираются стоять на ногах весь матч. Футбольные фанаты города Б. весьма законопослушны и не собираются покидать того места, на которое у них приобретён билет.

Добрая продавщица билетов Валентина хочет продать билеты фанатам таким образом, чтобы все они могли наблюдать за матчем и не мешали друг другу. Опишем, какие условия должны для этого выполняться.

Пронумеруем все ряды кресел от 1 до N , начиная с самого нижнего. Тогда фанат с ростом h_1 в ряду i не будет мешать фанату с ростом h_2 в ряду j , если выполняется одно из двух следующих условий:

1. $i > j$;
2. $i + h_1 < j + h_2$.

Работать продавщицей очень скучно, поэтому Валентине стало интересно, сколько существует различных способов продать билеты фанатам, чтобы они все присутствовали на матче, но ни один из них не мешал другому. Разумеется, Валентина не может продать более одного билета на одно место или продать билеты «без места». Но количество вариантов может быть слишком велико, чтобы посчитать их без помощи компьютера, а на то, чтобы написать программу для подсчёта, у Валентины не хватает квалификации. Валентина считает два варианта продажи билетов различными, если в одном из них хотя бы один фанат стоит не на том же месте, что в другом варианте. Известно, что Валентину устроит остаток от деления результата на 1 000 200 013. Удовлетворите любопытство Валентины, напишите требуемую программу!

Input

В первой строке ввода заданы два целых числа N и K ($1 \leq N \leq 1000$, $1 \leq K \leq 10$, $K \leq N$) — количество рядов и фанатов соответственно. В следующей строке через пробел записаны K целых чисел: i -е число задаёт рост h_i ($1 \leq h_i \leq 1000$) i -го фаната.

Output

Выведите одно число — остаток от деления на 1 000 200 013 количества способов продать билеты фанатам так, чтобы все они присутствовали на матче и не мешали друг другу.

Examples

standard input	standard output
3 2 1 2	4
5 1 4	5
5 2 4 2	13

Problem E. Столбики монет

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 512 мегабайт

Некоторое количество золотых и серебряных монет одного размера собрано на столе в n столбиков. Столбики могут иметь различную высоту.

За одно действие мы можем поменять местами две монеты, принадлежащие к различным столбикам, в том случае, если они находятся на одной высоте относительно стола.

Назовём столбик *одноцветным*, если он состоит из монет одного цвета (только серебряных или только золотых). Вычислите, какое наибольшее количество одноцветных столбиков можно собрать, сделав произвольное количество вышеописанных действий.

Input

Первая строка входа содержит одно целое число n ($1 \leq n \leq 1\,000\,000$) — количество столбиков на столе.

Каждая из последующих n строк состоит из букв «G» и «S» и описывает один столбик. Буква «G» обозначает золотую монету, буква «S» — серебряную. Монеты перечислены снизу вверх — от поверхности стола к верхней монете в столбике.

Общее количество монет на столе не превышает 1 000 000.

Output

Выведите одно целое число — наибольшее количество одноцветных столбиков, которое может быть собрано из исходной позиции с помощью применения описанных в условии действий.

Examples

standard input	standard output
5 GS SG SSG SGS SSSSG	4
3 GGG GG G	3

Problem F. Дорожный вопрос

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 512 мегабайт

Дорожная сеть Байтруссии пришла в упадок: по каждой дороге можно проехать только в одном направлении, одни и те же два города могут соединять несколько дорог, а между другими двумя городами вообще может не быть пути по дорогам...

Чтобы спасти положение, парламент Байтруссии издал закон, согласно которому за въезд в каждый город путешественник обязан заплатить налог. Величина налога фиксирована для каждого города и определяется бургомистром.

Закон вызвал недовольство торговых гильдий, которые обратились к королю с петицией. В результате король своим указом запретил требовать налог за въезд в город X с путешественника, который уже заплатил налог в некотором городе Y , в который можно добраться из X напрямую или через некоторые другие города (не обязательно те, через которые тот уже прошёл). Тем не менее, в этой ситуации путешественник может по собственной инициативе заплатить налог в городе X , если сочтёт это для себя выгодным. Платить налог можно только в момент въезда в город.

Купцу первой гильдии Биту Битычу Байтоградскому нужно добраться из столицы Байтруссии M . в крупный портовый город C . При этом купец, естественно, хочет заплатить как можно меньшую сумму в качестве налогов. Обратите внимание, что Бит уже находится в городе M , а значит, не обязан — и не может — заплатить в нём налог до начала движения.

По информации о карте дорог Байтруссии и величине въездного налога для каждого города проложите оптимальный маршрут для купца. Гарантируется, что путь из M . в C . по дорогам Байтруссии существует.

Input

В первой строке ввода записано два целых числа n и m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 500\,000$) — количество городов и дорог в Байтруссии. Во второй строке содержится n целых чисел C_i ($0 \leq C_i \leq 10\,000$) — величины налогов за въезд в города. Далее в m строках записано по два целых числа x_i и y_i ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$), описывающих i -ю дорогу, идущую из города x_i в город y_i . Между парой городов может быть несколько дорог. Для удобства город M . имеет номер 1, а город C . — номер n . Гарантируется, что из города M . можно добраться в город C . Числа в строках разделяются одиночными пробелами.

Output

Выведите единственное число — минимальную сумму налогов, которую нужно заплатить на пути из M . в C .

Examples

standard input	standard output
3 2 1 1 1 1 3 1 3	1
4 4 1 2 3 4 1 2 2 3 3 2 3 4	6
2 2 1 2 1 2 2 1	2

Problem G. Stickers

Input file: *standard input*
Output file: *standard output*
Time limit: 5 секунд
Memory limit: 512 мегабайт

В магазине за углом продаётся n различных видов стикеров, i -й вид содержит d -буквенное слово s_i , состоящее из заглавных латинских букв. Все стикеры одного вида идентичны. Можно купить сколько угодно стикеров каждого вида. Требуется составить слово t из продающихся в магазине стикеров, при этом разрешается наклеивать стикер поверх уже существующих. Запрещается отрезать части стикеров, а также как-либо ещё модифицировать их.

Более формально, изначально у нас есть строка z , состоящая из m символов «#» (здесь m — длина строки t). За одно действие мы можем выбрать любую из строк s_i и целое число p между 0 и $m - d$, и для каждого $j \in \{1, \dots, d\}$ заменить $(p + j)$ -й символ слова z на j -й символ слова s_i (символы в строках пронумерованы с единицы).

Требуется найти минимальное количество таких действий, после которого получится требуемое слово t .

Input

Первая строка ввода содержит два целых числа n и d ($1 \leq n, d \leq 50$) — количество видов стикеров и количество букв, написанных на каждом стикере.

i -я из последующих n строк задаёт один из видов стикеров и содержит ровно d заглавных латинских букв — слово s_i , написанное на стикере i -го вида.

Последняя строка ввода задаёт требуемое слово t , состоящее из не менее чем одной и не более чем 10^4 заглавных латинских букв.

Output

Выведите одно целое число — минимальное количество стикеров, которое потребуется для того, чтобы составить слово t . Если составить слово из данного набора стикеров невозможно, выведите слово «NO» вместо числа.

Examples

standard input	standard output
2 3 ABA BCB ABACABA	3
2 4 ABBA BCBA ABBCBBA	NO